Extra Credit Project

A File Input / Output and Text Processing Challenge

General Instructions:

**This project is for extra credit only, and is worth the same number of points as one lab assignment.** Your mission, should you choose to accept it, is to try your hand at opening a file and doing simple text processing.

1. Look at the slides for 4/21/2016, the slide where I instructed you to download an E-Book from Project Gutenberg. (You can reach Project Gutenberg at <https://www.gutenberg.org/>.) Download an E-Book in UTF-8 format and open it in a text editor. In Windows, a standard text editor is Notepad. Notice how there is a header, body text, and a footer. Note how words are generally separated. Think about what you are seeing.
2. Once you feel comfortable with the layout of a UTF-8 E-book from Project Gutenberg, go and design a **word** class using the object oriented programming techniques we have been studying this semester. Your word class should minimally correspond to the following UML, although you can explore doing more than this:

**Word**

* Type
* Frequency

+ getType()

+ getFrequency()

+ setFrequency()

1. Now, we can get to brass tacks. You’re going to write a program in C++ that reads in the body of a Gutenberg E-book and represents it as a vector of **word** objects. For every word in the e-book, create a word object for it, then add that word to the vector. Then, read the **Note:** The main challenge here is likely not to be representing the word, but distinguishing between **type** and **token**. Let’s say the word “the” occurs 10,000 times. Every one of those 10,000 occurrences is a **token** for the **type** “the”. You want to represent each type in the vector, not every token. There are many ways to do this. Hints: The skills you already have accumulated will allow you to insert the words in sorted order. This will allow you to search fairly quickly through your growing vector of words every time you retrieve a word from the file. If you’ve seen it before, increment its frequency. If you haven’t, set its frequency to 1 and insert it at the appropriate point. I’ll discuss a search algorithm called binary search in class that will allow you to do this.
2. **To get the extra credit, you just have one more thing to do.** Use your vector of types to do some simple, statistical analysis of your chosen e-book’s style. Write a program that will look through your vector of words answering the question: what are the top ten words by frequency? How many words occur only once? What word or words occupy the median, so to speak?

Good luck!